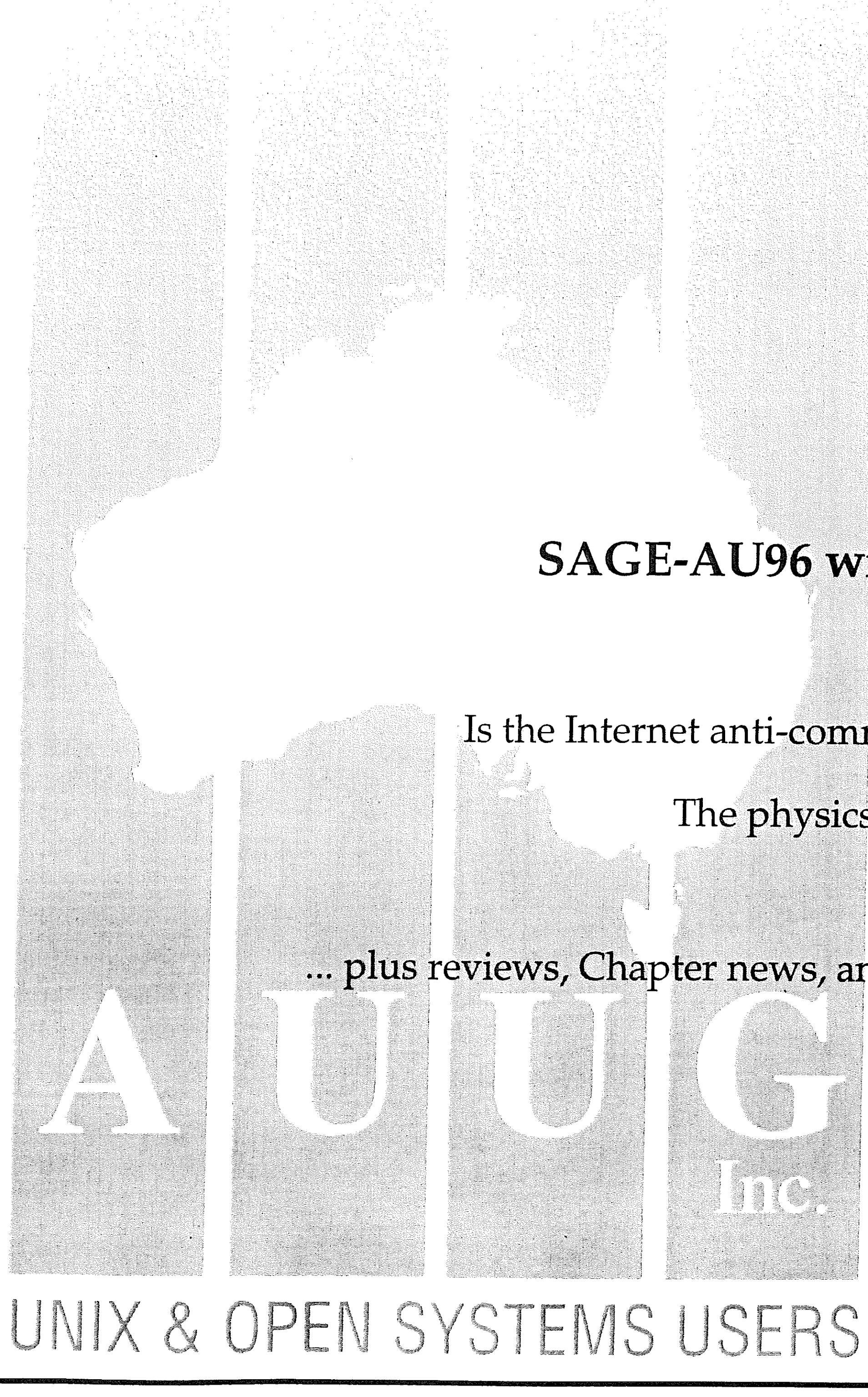
AUUGN

The Journal of AUUG Inc.

Volume 17, Number 4
August 1996



SAGE-AU96 wrap up

Is the Internet anti-commercial?

The physics of Java

... plus reviews, Chapter news, and more

ISSN 1035-7521

Print post approved by Australia Post - PP2391500002

AUUGN
Volume 17, Number 4
August 1996

AUUG Membership & General Correspondence

The AUUG Secretary
PO Box 366
Kensington NSW 2033

Tel: (02) 361 5994
Fax: (02) 332 4066
Freephone: 1-800-625-655
E-mail: *auug@auug.org.au*

AUUG Executive

President: Michael Paddon
mwp@aba.net.au
Kodak
173 Elizabeth St.
Coburg VIC 3058

Vice President: Glenn Huxtable
glenn@fs.com.au
Functional Software
PO Box 192
Leederville WA 6903

Secretary: Peter Wishart
pjw@auug.org.au
EASAMS/GEC Marconi Systems
PO Box 4806
Unit 7, 10 Kennedy St.
Kingston ACT 2604

Treasurer: Stephen Boucher
stephen@mtiame.mtia.oz.au
MTIA
509 St. Kilda Road
Melbourne VIC 3004

Committee Members:

Phil McCrea (Past President:)
pmc@syd.dit.csiro.au
Division of Information Technology
CSIRO
Building E6B
Macquarie University NSW 2113

Frank Crawford
frank@atom.ansto.gov.au
ANSTO
Private Mail Bag 1
Menai NSW 2234

Lucy Chubb
lucyc@sw.oz.au
Softway Pty. Ltd.
PO Box 305
Strawberry Hills NSW 2021

Chris Maltby
chris@sw.oz.au
Softway Pty. Ltd.
PO Box 305
Strawberry Hills NSW 2021

David Purdue
David.Purdue@aus.sun.com
SunSoft
119 Willoughby Rd.
Crows Nest NSW 2065

AUUGN Business Manager

Elizabeth Egan
auug@auug.org.au

Table of Contents

AUUG Inc. acknowledges the generous support of its corporate sponsors...

Contribution deadlines for AUUGN in 1996

Vol 17,#5 (October '96): September 20

Vol 17, #6 (December '96): November 22

PO Box 366
Kensington NSW 2033

AUUGN Correspondence

Please send all correspondence regarding AUUGN to:

AUUGN Editor
PO Box 366
Kensington NSW 2033

E-mail: *auugn@auug.org.au*

Submission Guidelines

Submission guidelines for AUUGN contributions are regularly posted on the **aus.org.auug** news group. They are also available from the AUUG World Wide Web site at **http://www.auug.org.au**

Alternately, e-mail to the above correspondence address, requesting a copy.

AUUGN Back-issues

A variety of back-issues of AUUGN are still available; for price and availability details, please contact the AUUG Secretariat, or write to:

AUUG Inc.
Back Issues Department
PO Box 366
Kensington NSW 2033
Australia

Conference Proceedings

A limited number of copies of the Conference Proceedings from previous AUUG Conferences are still available, at $50 each for members, and $60 for non-members. Contact the AUUG Secretariat for details.

Mailing Lists

Inquiries regarding the purchase of the AUUGN mailing list should be directed to the AUUG Secretariat. Telephone (02) 361 5994 during business hours, or Fax (02) 332 4066.

Disclaimer

Opinions expressed by the authors and reviewers are not necessarily those of AUUG Incorporated, its Journal, or its editorial committee.

Copyright Information

President's column

Michael Paddon

AUUG 96 has been and gone, and I suspect that all my fellow conference organisers are feeling both the same sense of relief (wow! I get my life back) and regret (but we were having sooooo much fun) that I am.

Those of you who could get to the conference this year will have noticed that it was a bit smaller and more focussed that the last few have been, although with an attendance of almost 500 it certainly wasn't a tiny event. This trend seems to have been quite healthy: the sessions were great, there was a renewed sense of community that went missing some time ago, and everyone looked to be having a lot of fun.

The conference network tied everything together, and was a testament to the capabilities of Stephen Boucher and the people from Access One. Just about every vendor in the trade show with a computer elected to connect, the speakers relied on it for their presentations, and the whole proceedings was broadcast on the MBONE. A network like this really sets our conference apart from all the wannabees, and I'd like to extend a big thank you to all the volunteers who put it together and made it work.

The unexpected highlight of the conference was the MPEG movie of the exploding whale. Yes, someone tracked down this famous piece of urban folklore and we managed to screen it at the AGM (who said they were boring?). Certain vendors in the trade show then got it going on their video displays, and the rest, as they say, was history.

If you haven't seen the exploding whale, don't despair. The MPEG will soon be downloadable from the AUUG web server for all to enjoy.

The conference dinner, as traditional, was rowdy and altogether too much fun. Balloon networking seemed to the order of the day, with just about every table connected by string held aloft by "repeater" balloons and bearing signs such as "BGP - Balloon Gateway Protocol" and "Sauce Routed Packet". Okay, I thought it was funny at the time...

Charles Sturt University provided their excellent wine for the evening, and Rob Kolstad entertained us with a quiz show. Yes, we had contestants vying for prizes in categories such as "RS-232", "Editors starting with the letter 'e'" and the like. Only at an AUUG conference...

There are a few people and groups whom I'd like to take this opportunity to single out for thanks. Without our sponsors it just wouldn't happen: thank you Access One, connect.com, NCR and Opentec Solutions.

Phil McCrea worked tirelessly to make the Internet Commerce Day a reality, with the help of sponsorship from the Australian Financial Review, CSIRO and IBM. The success of his vision was remarkable. Thanks, Phil.

Enno Davids and Lucy Chubb took a loose bunch of volunteers and managed to forge them into a team (this is akin to herding cats). How they did this, I'll never know, but we are grateful for their hard work.

Liz Egan, as AUUG's full timer, acted as the focus for our entire effort. She also brought her camera to the dinner, but we'll forgive her. Wael Foda was the dynamo behind the conference and exhibition management, and I'm sure you'll agree that his professionalism showed in the results.

Many volunteers donated their time and energy to our cause, on committees and on the conference floor. We also had enormous help from our friends at CSU. I guess it's pretty obvious where we'd be without your enthusiasm.

Finally, a big thank you to our presenters and to the people who came to hear them. We hope you got as much out of it as we did, and we hope to see you next year.

This is the last AUUG conference which will be co-organised with Charles Sturt University. We believe we have gained an enormous benefit by renewing AUUG's traditional ties with academia, and we hope that CSU has enjoyed the partnership as much as we have. Next year, however, CSU will be deeply involved in the International WWW Conference that is visiting our shores, and AUUG will be pursuing our trend of focussing back on open systems.

There has been no doubt that the Internet and the WWW have dominated AUUG's conferences these past few years. This was appropriate given UNIX's role in the growth of the global network and the amount of interest everyone had in these technologies. However, there is a lot of other interesting activity going on in the open systems world, and we want to make sure that these developments get fair airtime within AUUG. This is what we mean by improving our focus... not so much reducing Internet related activities but encouraging others.

The winter conference also brings us our AGM. We restructured our agenda and increased our scheduled time this year in order to encourage more of a question and answer format as opposed to the somewhat dry reports. This was quite successful, with a very active membership asking questions both easy and hard.

I think everyone was satisfied with the results, though I am interested in any comments you may have. The minutes, of course, will be published in due course.

This is my chance, as always, to point out that you don't have to wait for the AGM (or the elections for that matter!) to give your management committee feedback or suggestions.

What now? We are currently putting together a roadshow for November which should get to the majority of states. In particular, if your chapter missed to one earlier this year, we guarantee we'll be in your neighbourhood this time around. We're looking forward to seeing you there.♦

AUUG Inc

UNIX® AND OPEN SYSTEMS USERS

CALL FOR PAPERS

Technical Conference Series 1997

The AUUG *Technical Conference Series* will be held Australia wide in every state and territory, between February and April 1997, with the exact dates for each location to be available shortly.

AUUG invites proposals for papers and tutorials relating to the technical aspects of Unix and Open systems.

Presentations may be formatted as tutorials or technical papers, and should be geared at an audience requiring in-depth knowledge.

Technical presentations should be approximately 45 minutes, including a 5-10 minute question time.

Tutorials, providing a more thorough presentation, should be either a half-day or a full-day in duration.

Submission Guidelines

Those proposing to present papers should submit an extended abstract (1-3 pages) and a brief biography.

Those submitting tutorial proposals should submit a plan for the tutorial and a brief biography.

Please indicate on your proposal whether it is a paper or tutorial, and for tutorials whether they are half day or full day.

Location

The conference series will be held Australia wide, so please indicate in which city you wish to present - however, you can do so in more than one!!

Speaker Incentives

Presenters of papers are afforded complimentary conference registration.

Tutorial Incentives

Tutorial presenters will receive either a flat fee or a negotiated percentage of their tutorial profit.

Proposals should be sent to:

AUUG Inc, Level 4, 90 Mount Street, NORTH SYDNEY NSW 2060

Email: auug@auug.org.au

http://www.auug.org.au

Java Enters new Evolutionary Phase

David Flanagan

The 6,000 recent attendees at JavaSoft's inaugural U.S. Java conference, JavaOne, confirm what was already becoming abundantly clear: Java has been accepted by programmers and industry alike as the network programming language of choice. Despite some bugs, platform inconsistencies, and lack of features in the 1.0 releases, it is plain: Java is here to stay.

With this acceptance, Java enters a new phase of its evolution. In recent months, there has been an explosion of new products and packages for Java, along with new acronyms, marketing slogans, and hype, of course. With all this activity, it has become difficult to even answer the question "What is Java?". A few months ago, we would have said that Java was a programming language, and a class library, and a virtual machine definition. This is still true, but now, to keep the class libraries all straight, we must resort to terms like the "Java Platform", and the "Java Base API".

With release 1.0.2 of the Java Developer's Kit, the Java language itself is stable. An internal debate is ongoing at JavaSoft as to whether a future version of the language should support method pointers and perhaps templates, but in any case, these are not changes that we will see in the short term.

It is on the API front that we will see the most rapid change. Four of the most important new APIs, and four that we'll see relatively soon are JDBC, RMI, IDL, and Java 2D. They are described in more detail below.

JDBC is the "Java Database Connectivity" API. It is a call-level (not embedded) SQL interface that allows Java applications to communicate with databases by executing SQL queries and statements and retrieving their results. The JDBC API was finalized and frozen on June 8th. In the near term, JavaSoft will provide a JDBC-ODBC "bridge"—code that allows Java JDBC code to communicate with existing ODBC drivers which communicate with the database. It won't be long before database providers and other vendors market JDBC drivers that communicate directly with databases. See `http://splash.javasoft.com/jdbc` for details about JDBC.

RMI stands for "Remote Method Invocation"—the object-oriented equivalent of RPC (Remote Procedure Call). The RMI API, and its associated object serialization API allow Java applications to communicate with each other by passing data values back and forth, and by invoking methods in remote objects. RMI and object serialization allow object references to be shared between applications, and also allow complete objects to be passed over the network. Related to RMI is the IDL API, an implementation of the CORBA-standard Interface Definition Language, which will allow Java applications to share objects with applications written in other languages. Alpha versions of both RMI and IDL are available from `http://java.sun.com/devcorner.html`.

Java 2D is a new graphics API that will greatly extend the current graphics capabilities of Java. It is a Java interface to the Bravo graphics engine from Adobe systems, the makers of the PostScript page description language. Bravo is the graphics engine that drives Adobe products such as Acrobat, PageMaker, and Illustrator. The Java 2D API is essentially a Java interface to an enhanced version of PostScript and will include very powerful drawing, imaging, and transformation features. The release date is not firm for the Java 2D API, but pre-release versions should be available relatively soon.♦

David Flanagan is author of the bestselling book "Java in a Nutshell", from O'Reilly & Associates.

The End of AUSCERT?

Frank Crawford

On the thirteenth June the Australian Internet community was shocked by the announcement that AUSCERT, the Australian Computer Emergency Response Team, would close down effective almost immediately, due to a funding crisis. For the average Internet user this may have not meant much, but for those involved in the operation of ISPs and Internet security, it was a major, and in many cases, unexpected blow.

The reason for the shock is that a major element in the infrastructure of the Australian Internet was suddenly being removed. AUSCERT has often been described as the "network police", but this is not really correct, rather they act as an information clearing house on network security matters. Further, as they are Australia's member of FIRST (the Forum of Incident Response and Security Teams) they keep in close contact with Internet security groups overseas, and keep a finger on the pulse of incidents world wide.

The major contact most groups have with AUSCERT is through their advisories, i.e., information bulletins on security issues. These advisories range for notification of known security problems and their fixes, to an analysis of the current incident and activities. They also publish such items as security checklists and lists of security related patches.

Another service that AUSCERT offers is an FTP site with a comprehensive collection of security packages and tools, documentation and other security related items. This has been collected and managed in conjunction with the other FIRST members.

Of more importance and yet less widely used (hopefully) service is to provide a single trusted point of contact for security incidents within Australia. This service operates 24 hours a day, 365 days a year, and allows a quick response to emergencies so as to minimise damage. In fact this is AUSCERT's most valuable service and the real reason that it must be maintained.

AUSCERT was originally formed after a number of such security incidents in the early days of AARNet. At this time there were a number of attempts to access computers in various tertiary institutes and research organisations (the predominant sites at the time). At the peak of these attacks, as there was no local coordinated response, a number of them were successful, causing the destruction of information and endangering expensive research equipment.

Since that time, the number of threats has grown dramatically, as has the cost of incidents. Yet at the time when there is the greatest need for AUSCERT, it is in danger of folding, due to lack of funding. AUSCERT was originally funded by the AVCC to cover AARNet, however, when Telstra took over AARNet that funding was only continued for a further year. That year is now up, and no further source of funding has been forthcoming, hence the shutdown.

In effect, this is like Telstra discontinuing the 000 service due to lack of funding. Any emergencies can still be handled, but not easily or quickly, and in fact, most people will not know what to do. For anyone involved in computer security this is a major problem, while for anyone involved in computer crime it is a major opportunity.

The crucial issue here is the future of funding for AUSCERT, the initial threat of shutdown had the effect of obtaining interim funding, but only until the 31st October. If further funding is not found by then, AUSCERT will shutdown permanently. The business case put forward by AUSCERT favoured government funding through some agency, as is the case for most other CERTs around the world. Unfortunately, this will not be the case in Australia, and AUSCERT must operate on a user pays model.

Over the next few months, AUSCERT will be working on obtaining sufficient commitment from the Australian Internet community for continued viable operation. If this funding isn't found, Australia will, at least for some time become a "happy hunting ground" for crackers. Such a condition couldn't continue forever, rather, eventually, an organisation with similar aims to AUSCERT would again be formed.

Given this, it would be much better to support AUSCERT at this stage and avoid the intervening trauma, which would have a major and disastrous effect on the growth of the Internet in Australia.♦

Internet Industry

Phil McCrea

On occasions people often use simple comparisons to make a point: you state something, and then compare it with its logical - or perhaps illogical - opposite. For instance, around the 17th of March, you often hear the expression: "There are two kinds of people in the world: those who are Irish, and those who would like to be Irish" - implying that no-one likes to be left out on St. Patrick's day. Being Irish born and bred, I feel a certain affinity to this statement!

In the programming world, I have often heard it said: "There are two kinds of programmers in the world: those who understand pointers in C, and those who don't". Experience C programmers will understand the subtlety of this.

Recently I've found myself uttering: "There are two kinds of people in the world: those who understand the potential of the Internet, and those who have yet to understand the potential of the Internet". I certainly belong to the former group: indeed I hold a passionate belief that the Internet will change the face of Australian industry like nothing we have ever experienced before. Its effect will be just as profound as the industrial revolution, or as the invention of the internal combustion engine.

So convinced am I of the changes that the Internet will bring, that I have difficulty understanding how anyone in the information industry particularly the

IT industry - can fail to grasp what the Internet offers. And yet I meet them all the time! When the Internet is mentioned - it's even dinner party conversation these days -often people's faces start to take on a quizzical look, and they utter things like "Yes, but how can anyone make money out of the Internet?".

My standard reply is along the lines of: "Well, who's making money out of the telephone?" There are 2 answers to the telephone question, of course firstly telecommunications carriers, and secondly people who use the telephone. The first group is easy to quantify - Telstra's $15B odd turnover is testimony to this. The second group contains just about everyone else in the country, including 12 million fixed line subscribers, and the 4 million mobile telephone users. The 12 million of us may not make money directly from having a telephone - but it sure helps! Imagine running a business without a telephone - or a fax for that matter.

It's important to understand that the Internet is basically a transport mechanism - remarkably fast, remarkably cheap, and oblivious of national boundaries. It is 'enabling technology' which makes lots of other things in business happen much more efficiently.

There are 4 main industry 'groups' in what can loosely be called the 'Internet industry'. Firstly, the telecommunications carriers who supply the 'pipes'. There are two of these at present - Telstra and Optus (although in reality Optus is a VAR for Telstra), but next year there will be several more after de-regulation takes effect. Clearly the carriers are making money out of the Internet.

The next level in the hierarchy are the Internet Service Providers (ISPs) of which there are around 200 in Australia - and increasing. Not too many of these organisations have gone under. In fact it is perhaps perceived as a glamour sector - look at the recent success of Ozemail's float on Wall Street.

The next level in the Internet hierarchy is the group of companies how provide the infrastructure for secure transactions over the Internet. This is a small number of organisations, but also growing. There are not many companies at this level who are making money yet, but it will not be long perhaps the end of the year - before the required encryption code is embedded in the popular Web browsers. At this stage, business from Internet transactions will become lucrative indeed.

The final group are the so-called 'content' providers. The business model here is generally subscription to a service, or sale of a service of product. Most organisations who operate in this space are still discovering the rules. But this will change, particularly with the advent of electronic cash, which will enable small purchases to be made over the Net - for instance buying a newspaper.

So who's making money out of the Net? At present, mainly the carriers and ISPs. In a year's time transactional revenues will become significant. With all this in place, the content providers will make a mint... ♦

The Physics of Java

Michael Paddon

In the physics of the open systems universe, the C programming language has served as a fundamental force ever since that continuum's equivalent of the big bang. Just like our own universe, it is hard to pinpoint the exact birth moment of "open systems", however it is common practice to date the epoch from the creation of UNIX (circa 1969-1970).

This is a little inaccurate to say the least, given that UNIX was written in assembler (and hence not particularly portable) and that it sported a very limited functionality. It was, however, a timesharing system from the first, which compares quite amusingly against the recent history of mainstream PC operating systems.

In fact it wasn't until First Edition (1971) that UNIX was written in a portable, albeit limited, systems language called "B". By Fourth Edition (1973), UNIX was written in C, and it was beginning to exhibit the portability that we consider its natural domain. Fifth Edition (1974) was made freely available to educational institutions and the era of open systems had clearly begun. Here ends the history lesson!

The point of all this was not to bore you to death with a dry history, but rather to illustrate that this thing we call "openness" (is that a real word?) is tied far more closely to the C language than it is to any operating system. This was true at the genesis, and remains true today with standards such as POSIX loosening our shackles to any one operating system. Even within the UNIX world there is wild variation, yet this diversity does not inhibit the porting of code that was written to be portable.

This extends to the non-UNIX system as well, assuming that the vendor approaches portability with an honest effort, and not just a "tick the box" mentality. Regular readers of AUUGN will recall last year's fuss over the Windows/NT POSIX subsystem as a case in point.

Which brings us full circle to my original metaphor. The C programming language is the gravity of the open systems universe, a constant force that binds all things open together. Just about every important piece of open software is written in C, or a derivative (like C++), or in a language whose compiler/interpreter is written in C.

Not bad for a flawed language. Remember what a truly dangerous tool C was before ANSI got to it? Even today, there are still warts left over from when it used to be a high level assembler, as well as more fundamental problems: for example consider issues as diverse as scoping, exception mechanisms and multilingual support. Even when compared with its contemporaries, Algol and Pascal, C looks somewhat anaemic.

But the strength of C does not lie in theory, it lies in practicality. What you can effectively achieve with a language is far more important than any other measure. Make note of this fact, for we will return to it.

Today there is a new game in town: Java. Cutting through all the marketing hype and propaganda (both positive and negative), Java is a relatively simple new object oriented programming language that assists in the construction of network downloadable applications. That's just about all there is to it, despite all those millions of words written about it by people who think that the Internet was invented in 1992, that HTML is a programming language, and that the phrase "paradigm shift" is a cool and edgy thing to say.

Let's explode a few myths and make a few observations.

There is nothing special about Java as a language. Sure, it does garbage collection, it's got references instead of pointers and it has a security model enforced by its virtual machine. None of these are ground breaking (though some of the security stuff is pretty interesting).

Similarly, C is not special as a language. If Dennis Ritchie had decided that an extended Algol was a great systems language we'd all be compiling with a GNU Algol compiler today (gal?).

Java, like C, is highly versatile and portable. In fact Java probably has an edge here, given the virtual machine technology it is layered over. Java also has a highly functional and stable set of libraries, which allows it to be an immediately useful tool for building applications and interfaces without every programmer needing to reinvent the wheel.

There's nothing special about Java's libraries, though, and countless examples of similar bodies of software are available for just about every modern language.

Java does create the opportunity for a completely novel class of application: the ubiquitous, downloadable applet. This is more of a function of the virtual machine however, and many other high level languages could be targeted at the same execution environment.

We could have been really unlucky and have had Gosling decide that Lisp was a spiffy high level language to base an applet language on. Hardly bears thinking about, does it?

Where does this leave us? Java is obviously a good, but far from perfect, language that is internet portable and allows us to construct new kinds of programmes. Not unlike C, Java is a pragmatic technology, in the right place at the right time.

As we have already seen, it is languages that drive a new technology, and it is likely that Java (or something like it) will be the powerhouse of the next wave of open systems; one based on downloadable applets and distributed computing services. Java may yet falter as the vehicle of this change, but it is clearly the prime contender. Somehow, Sun have convinced just about everyone to run the Java virtual machine, which just goes to show the benefit of being first in a marketplace.

In any case, I am dead sure that it is a *language* and not a browser or a web server or an operating system or even a database that will be the keystone of future global computing. I'm backing Java. What do you think? ♦

A Wrapup of SAGE-AU96

Frank Crawford

Another annual conference for The Systems Administrators Guild of Australia (SAGE-AU) has gone, and again it proved to be both popular with and important to system and network administrators throughout the country. Systems Administrators are the people responsible for the management and operation of computer systems in most organisations.

While SAGE-AU is not specific to any operating system, the dominance of UNIX systems in the server market means that most SAGE-AU members either administer or have previously administered UNIX systems. On the other hand, most members have an interest in Microsoft NT systems, as something they will be involved with at some stage.

The 4th Annual Conference, SAGE-AU'96, was held in Brisbane, at the University of Queensland, from 16th - 19th July. The conference started with two days of tutorials, covering such topics as Advanced Administration topics, Java Programming and An Introduction to PGP (Pretty Good Privacy). These were very well attended and gave administrators a chance to catch up on some of the latest techniques or systems.

The conference proper was held over the two days of the 18th and 19th, and consisted of a number of papers and work-in-program sessions, presented by administrators from throughout Australia. Again, the topics covered ranged from Automating Administration Tasks, through to case studies of various commercial and academic sites, and on to network and system security issues. These case studies included the management of a large FTP site, implementation of a site monitoring system, planning a major site move and the introduction of NT at a major commercial site.

The presenters included Craig Bishop, the Technical Manager of connect.com.au, Eric Halil from AUSCERT, and Peter Elford from CISCO.

As well, Dinah McNutt was the invited international speaker. Dinah has strong connections with USENIX/SAGE, having been the chair of the 1995 USENIX LISA conference. She has also written for Byte, SunExpert, iX Magazine and has a regular column in Unix Review Magazine.

Aside from general administration issues, the major issues coming from the conference were either security related or management issues of servers in a large network. In terms of security, there were numerous mentions of PGP for both encryption and authentication. One of the big issues with PGP and similar public key encryption schemes, is the verification that the public key actually belongs to who it is claimed to. This is slowly being addressed by the establishment of Certification Authorities, and this was described in a presentation by Catherine Allen.

The use of encryption and authentication within a local network was also a major issue, with numerous references to "ssh", a remote shell replacement, which implements both public key encryption and an encrypted channel between systems. When this is teamed with "rdist", a program to automate software (and other) distribution, to become "sdist", it provides a secure means of administering a large number of systems. A plan for using these tools to form a secure environment was presented by Geoff Halprin, and should form the basis for many organisations.

Arguably, the most important presentation of the conference was by Jim Hogan and Peter James entitled "On-line regulations and System Administration", which covered many of the regulations proposed throughout Australia. An outline of the issues which may affect all ISP and the administrators that manage their systems was given, including a summary of the difference across the country, convinced most people present that there is a long way to go to get sensible laws, but that we are now moving in the right direction.

Ultimately, the conference proved useful to all those who attended. It gave them information that was immediately useful, an understanding of some of the principles underlying their activities, and an indication of some of the forces that will affect their future.

Further information about SAGE-AU can be obtained by contacting Frank Crawford, the secretary, on (02) 717 3015 or email `secretary@sage-au.org.au`.♦

The Network Computer

Phil McCrea

The Network Computer (NC) has had a fair amount of airing recently, fuelled by speculation from Oracle supremo, Larry Ellison. But while several hardware manufacturers have announced that they will produce NCs for around $500, the NC is still largely speculation at present.

It's easy to see why Larry Ellison is so gung-ho on NCs: Oracle's strengths lie in the server end of the market, while Microsoft's strengths are in the client, or PC, end. Larry wants the 'business side' of things to live on the server; Bill wants everything to happen on the PC where he is in control. Egos are at stake here...

On the face of it, you would have to give Larry good odds that the NC will become a reality. If you look at the trends over the past few years, just about everything in the computing arena is becoming network-centric. The fatter the communications pipes become, the more useful networks will be, and the more we will be able to access information that resides elsewhere.

So, will the NC take over from the PC? The answer in a qualified No, because the PC and the NC address two different markets. Yes, the NC will become popular in the near future. It will be diskless , and will be used as an 'information terminal'. NCs will be aimed at the mass market: they will be sold by companies that sell car radios and burglar alarms. They will become commodity items, with household brand names. They may look a little different to a PC. They may be sold without a screen, with the intention of being used with a TV monitor - there are already a couple of these on the market, complete with a remote control adorned with 'TV' and 'Web' buttons. In the near future, they will probably be sold with flat panel displays that will be wall mounted.

So what would you use an NC for? Basically for information retrieval of all sorts: telephone numbers, weather, sports results, shopping, entertainment information, etc., and of course for electronic commerce. Mainly the kind of things you would currently look up in a newspaper, or in a catalogue. It will soon be second nature to 'check the NC' - much like our fingers currently go walking!

The NC will not be a cumbersome beast like a PC: there will be nothing mechanical to go wrong - in fact it will probably be constructed from a chip or two. It will not take a couple of minutes to boot like current PCs do; nor will it spew out silly error messages. Because it will draw next to no power, it can be left on all of the time. Even when first powered up, it will boot itself over the network in a matter of seconds, and be at your service. In short it will be like a telephone: in fact it may very well be integrated with a telephone handset.

Some NCs will operate without a keyboard, with people making choices my moving arrows around a screen. I believe most people will want to use a keyboard, however: moving through a series of screens with lots of cursor movement to finally convey a message such as 'I want to book a hotel in Port Macquarie' could be fairly frustrating!

What about applications where you may want to store things - like word processors and spread sheets? Will users trust some third party to look after their confidential information? Microsoft came to grief with this approach when it tried to set up a global network ;last year, where everything was to reside on servers at Redmond! NCs may be different, however, as second generation NCs will probably have a smart card slot which will contain your personal encryption key(s), so you will be able to encrypt your information and store it safely a remote server.

And what of the PC? Well, there will always be a market for people who want to do all their work on their own machine, and store it locally. There will be 'power users', desktop publishers and software developers who will need local facilities such as CD players, specialised plug-ins, and the like..

But by far the majority of users will want Web access most of the time purely for information, and in this case an NC is all that is required.

For those who want to carry out word processing or use spreadsheets, then an NC with a decent screen, keyboard, printer, and smartcard security may be all that is necessary.

Perhaps Mr. Ellison may get his way after all... ♦

Is the Internet Anti-Commercial?

Frank Crawford

With the rise of more commercial companies connecting to the "net", there is a growing backlash against the "commercialisation of the net". In particular there are some groups that are vigorously protesting about this commercialism, and the loss of the nets "innocence".

As a long-time user of the Internet, it is worth stepping back and looking at what this commercialism means, is it good or bad, or just different. Most users are not really anti-business, however, most are against the inappropriate use of the net.

Many businesses have grasped that the net is a new marketing medium, but have not understood that the old techniques are not appropriate. For example, mass mailout may work through the postal system, but in an electronic system, similar activities (aka spamming) don't. While there are a number of reasons for this, probably the biggest is also the reason organisations try them, they are quick and simple. However, when compared to normal mail, how many bits of junk mail can fit in a physical letterbox, and how many in an electronic one? Also, unlike normal junk mail, the receiver can easily fight back, as it is simple to send back the junk, causing the original senders system to "meltdown".

In fact the basic problem here is that there is currently no real way to prioritise email, or to handle different classes. For example, how do you put up a "No Junk Mail Here" sign?

What this illustrates is not that the net can't be used for business, but rather that new methods have to be developed. This is a far wider area than just how to do payments securely over the net.

Most people using the net originally were after free information. Today, most are after relevant information, and are willing to pay for it. The only catch here is that they want a simple and safe (in that order) mechanism for handling the payment.

The other major criterion for this information is that is must be both rapid and accurate, be it bug fixes for their code, or information on their favourite rock group. Companies that can do this are generally seen as good net citizens, whereas those that don't or won't are not. Of course, very few businesses have really come to terms with the net and the expected response time, so there are currently far more "bad citizens" than "good citizens".

Given that the web is barely two years old, and most companies set up 5 year plans, it is not surprising that many are just not capable of moving fast enough. Over time, when the pace of both growth and change slow down (as it must eventually) many of them will catch up. As an example, take a look at Microsoft's recent change in attitude concerning the net.

The other side of this is that the information must be accurate. Traditional PR blurbs that "colour the truth" quickly come under fire as (to quote a popular TV program) "the truth is out there" and is soon discovered. Once this happens both the reputation of the company and the integrity of any previous (or future) statements is compromised.

Many of the complaints about business activities are counter productive. If many of these people had their way, we would still have a 2400 baud UUCP connect to the USA, as opposed to a 32Mb IP link.

Unfortunately many of the complaints fall into the category of "things aren't as good as they were in the past", but then that is true in most of life.

Objecting to business on the net is like complaining about advertising on TV, it may feel good, but it is fruitless. Businesses have sufficient money and power to move things in the direction they want, even if in many cases they don't know what that direction is. It is far better to guide them into a path that is acceptable to both the business and the net user (their customers) as it will be advantageous to both.

In fact, just about everyone will have some interaction with a business over the net. This can range from email to a business associate, to looking up book or CD catalogues, to direct purchases over the net, and so will benefit by the increased commerce.

In the end what will be found is that such commercialism is neither good nor bad, but just different. It will mean that the net is no longer the simple free place it once was, but it will also open up far more opportunities than before. Both the problems and the possibilities increase as the net grows.♦

Do Non-Technical Managers Work?

Frank Crawford

In just about every computer related organisation (and many others as well) there are two disparate groups, the managers and the technical staff. There is very little migration between the two groups, and almost all of that being from the technical to the management staff, primarily because it is one of the few means of advancement within an organisation. Generally, those in the management group have little understanding of technical issues, being willing to leave such details to others.

At the same time more and more large projects are said to be running into trouble, being canceled or otherwise failing. We are also seeing more major "non-commercial" projects being managed from start to finish by technical groups working on a part-time basis.

The Internet is a prime example of this, beginning its life as a research project, almost totally designed and used by technical types. It achieved its current position through people with technical knowledge and skills identifying and correcting problems as they were found, rather than directing their efforts into the "management" of the problem.

Even further, the biggest impetus to the growth of the Internet, and it's conversion to a "network for everyone", was the development of the "World-Wide-Web" and in particular the Web browser, Mosaic. Again this was a strictly technical project, developed more because the facilities were available, than because of a perceived need.

From these examples you may conclude that technical staff are only good at initiating new projects, but once it goes into "production" it needs to be "managed". Again, there are numerous cases of projects being taken through various cycles by technical people.

Again, the Internet is another fine example of this. As it continues to evolve and grow, each technical challenge is being met and addressed by those with knowledge and understanding of the technical issues.

Arguably one of the best example of this is Linux, which, through a world-wide effort, has been developed from scratch, and is continuing to develop and improve. Prior to the release of Linux 2.0, there had been a development effort that culminated in this release.

Since then the pace of development has increased, with 12 minor releases occurring in its first two months, each of which has increased the stability of the original version. The organisation (or management) of these releases is itself a major project, and has progressed without any noticeable problems.

The management of these projects have a number of things in common, which is generally not taken into account, or even understood, by those without a technical background. Firstly, and obviously, there is a requirement for understanding of the issues involved, much of which can only come about through a technical knowledge.

Secondly, and probably more important, all those involved in the projects have a commitment and feel a responsibility to ensure that the project succeeds. Unfortunately, many commercial projects remove these incentives from the technical staff, and yet expect them to perform.

Working in a technical role has a number of other benefits, one of which is the ability to see progress in the work. In most cases managers have to rely on arbitrary milestones and goals rather than a concrete result. Given this often arbitrary nature of these goals, it is difficult for them to see or feel where a project is up to.

Like anything else, these statements are generalisations. There are managers who can control big projects, without understanding the technical details, but they have a good feel for their people. However, there are far more non-technical managers, who do not have any such feel for their people, and add considerable extra pressure to their technical staff, often overwhelming any of their efforts to succeed.♦

AUUG96 Conference & Exhibition

Frank Crawford

Well, another year rolls around, and the Premier Open Systems conference, AUUG96, draw upon us. This year AUUG is holding the conference jointly with Charles Sturt University's Asia Pacific World Wide Web Conference, making it the biggest UNIX, Open Systems and Web event of the year.

Following last year's format, the conference, which is being held in Melbourne between 18th and 20th September, will be preceded by two days of tutorials and workshops (16th & 17th). These tutorials, which are either a full day or half day, allow an indepth coverage of such topics as Cryptography, SGML, PERL, Java, NIS+ administration, PGP, security of WWW and network management issues. As well, for those not into such technical aspects, there are also tutorials on "An Introduction to the Web", "Art History Resources on the Internet" and "Network Publishing".

If the topics are not a sufficient drawcard, many of the presenters are internationally recognised in their fields. These include people like Rob Kolstad from Berkeley Software Design, Greg Rose, the co-moderator of sci.crypt.research, Peter Elford from CISCO, David Purdue from SunSoft and Mike Greenhalgh, Professor of Art History at ANU.

While the tutorials are aimed at giving attendees an indepth understanding of some topics, the conference will give the participants an overview of the latest in information security, publishing on the net and Internet commerce.

As in previous years the conference will begin each day with a keynote presentation from acknowledged experts, then breakup into multiple streams covering selected areas, and finally rejoin for a final presentation.

Some of the speakers include Bill Cheswick from Bell Laboratories who will be talking about Internet Security and the World Wide Web, Roger Fraumann, from NCR and a familiar face to AUUG, whose presentation is entitled "Digital Dial Tone - Electronic Commerce on Demand" and Rob Kolstad with "The Future of the Internet".

However, these keynote speakers are not the only attraction, as this conference is also a chance to hear from those working in the trenches, and to get first hand what is really happening in the UNIX and Internet world. Some of the topics being covered include security, from both a development and management perspective, support tools, Linux, FreeBSD and other Public Domain Operating Systems, Java, Networking and Data Archives. As well, the Management stream will include case studies of various organisations.

For those who find they cannot afford the time to attend the entire conference, this year we are running (as a part of the conference) a special Internet Commerce Day on the 19th. This program has been put together in a manner which can be attended individually, but still fits with the rest of the conference, and includes such people as Bruce Wilson of CyberCash Inc, Randy Whitting from CommerceNet and Rob Morel of Oracle.

Finally, what would the annual AUUG conference be without one of the biggest exhibitions of Open Systems technology in Australia. As in previous years, there will be an exhibition running from the 18th to the 20th, showcasing products from such companies as Access One, OpenTec Solutions, NCR, Sun Microsystems, Silicon Graphics, SAS Institute, Megatec, Digital Equipment Corporation, Tecktronix Australia, and many other major Open Systems and Internet vendors throughout Australia.

This provides one of the few places to see all the latest computing systems in the one place.♦

THE BEST TOOL FOR PUBLISHING ON THE WEB?

IT'S THE O'REILLY BOOK NEXT TO YOUR COMPUTER.

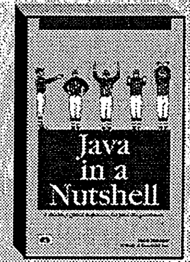
We write books that help you master Internet publishing and programming topics, going above and beyond mere mechanics. Our books explore subjects in-depth and are written and reviewed by experts. Next time you encounter a problem, reach for the solution that doesn't require bits or bytes or RAM. Reach for an O'Reilly book.

WEB CLIENT PROGRAMMING WITH PERL
By Clinton Wong
ISBN 1-56592-214-X, $59.95
Due Date: 12/96

Teaches you the basics of how browsers communicate with servers and how to write your own customised Web clients to automate common tasks.

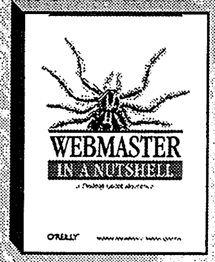
PROGRAMMING PERL, 2ND EDITION
By Larry Wall, Tom Christiansen & Randal L. Schwartz
ISBN 1-56592-149-6, $79.95

The authoritative guide to Perl—the scripting utility now established as the World Wide Web programming tool of choice. The book is coauthored by Larry Wall, the creator of Perl.



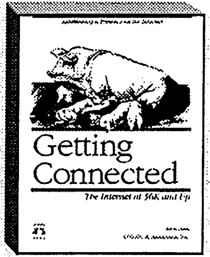
JAVA IN A NUTSHELL
By David Flanagan
ISBN 1-56592-183-6, $39.95

This comprehensive volume contains descriptions of all the Java classes and their related calls and introduction to important Java concepts.



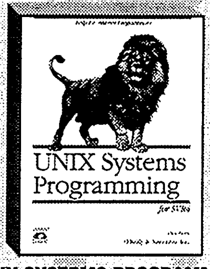
WEBMASTER IN A NUTSHELL
By Stephen Spainhour & Valerie Quercia
ISBN 1-56592-229-8, $39.95

This quick reference covers HTML, CGI, Perl, HTTP, server configuration, and tools for Web administration.



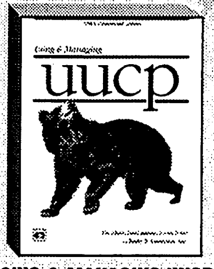
GETTING CONNECTED
By Kevin Dowd
ISBN 1-56592-203-4, $59.95

A complete guide for businesses, schools, and other organisations who want to connect their computers to the Internet. Covers everything you need to know, from helping you figure out which services you really need, to providing down-to-earth explanations of telecommunication options.



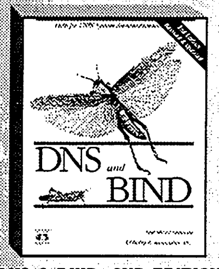
UNIX SYSTEMS PROGRAMMING FOR SVR4
By David A. Curry
ISBN 1-56592-163-1, $69.95

A comprehensive look at the nitty gritty details on how UNIX interacts with applications. If you're writing an application from scratch, or if you're porting an application to any System V.4 platform, you need this book.



USING & MANAGING UUCP
By Ed Ravin, Tim O'Reilly
Dale Dougherty & Grace Todino
ISBN 1-56592-153-4, $59.95

Using & Managing UUCP describes, in one volume, this popular communications and file transfer program. UUCP is very attractive to computer users with limited resources, a small machine, and a dial-up connection.



DNS & BIND, 2ND EDITION
By Paul Albitz & Cricket Liu
ISBN 1-56592-236-0, $65.00
Due Date: 1/97

This book is a complete guide to the Internet's Domain Name System and the Berkeley Internet Name Domain software. This second edition covers Bind 4.8.3, which is included in most vendor implementations today, as well as Bind 4.9.3, the potential future standard.

O'REILLY™

DISTRIBUTED IN AUSTRALIA BY WOODSLANE PTY LTD

Phone (02) 9970 5111 Fax (02) 9970 5002 Email techbooks@woodslane.com.au
PO Box 935 Mona Vale NSW 2103

UNIX Tricks & Traps

Subeditor: Gunther Feuereisen <gunther@agsm.unsw.edu.au>
Tel: 02 9931 9314   Fax: 02 313 7279

Hi everyone! Here we are in sunny Sydney, having just survived the monsoon-like rains (which incidently, are great for washing a car).

Alas, no contributions from anyone out there (if you're reading this drop me a note and say "Hi!").

As always, contributions are welcome!

Aloha!

TR(1) - MORE THAN JUST A TWO LETTER ACRONYM.

Ever copied a whole bunch of DOS files to your disk, and then become annoyed

with all those CAPS? Let's face it, lowercase is a lot less loud `;-)` Give `tr(1)` a gander - it won't immediately solve the problem, but with a little thought (and a few clickety-clicks) you'll have bashed out a nice little script to take care of those nasty DOS filenames.

For those of you who don't feel like clickety-clicking, this is the sort of thing I mean:

```
#!/bin/ksh
# swc - switch case
#      A Shell script to toggle file name case
usage="Usage: swc { -u | -l } file [ file ... ]
      -u      toggle to upper case
      -l      toggle to lower case
      file    filename or wildcards"
case $# in
0)     echo "$usage"; exit 1;;
*)
      type=$1
      shift
      case $# in
      0)      echo "$usage"; exit 1;;
      esac;;
esac
for file in `ls -d $*`
do
      case $type in
-l)     mv $file `echo $file | tr '[A-Z]' '[a-z]'`;; -u)     mv $file `echo
$file | tr '[a-z]' '[A-Z]'`;; esac
done
```

HOW TO MOVE OR COPY LARGE VOLUMES OF DATA - THE TAR(1) COMMAND.

One thing that often crops up is moving large amounts of data. Often across filesystems, or even within your own home directory. Now `cp(1)` will do the job quite nicely, but in many cases, you have set unique file/directory permissions on certain files, and you'd like those to be maintained. This is where `tar(1)` comes into its own.

Now if you take a look at the `tar(1)` man page, the first thing you'll see is something like this:

> *"The tar command saves and restores multiple files on a single file (usually a magnetic tape, but it can be any file)."*

The lovely thing about a system where *everything* is a file: you write something to work on files, and suddenly it works on devices too, and vice versa.

You will also find a multitude of options - I've never used more than three flags at any one time [1], and I remember a total of five flags. The syntax for `tar(1)` (not the complete syntax, look at your man page for that, this just includes the flags I use) is as follows:

```
tar { -c | -x | -t } [ -v ] [ -f Archive ] [ File | Directory ]
```

In "English" the flags mean:

-c	Create archive
-x	eXtract archive
-t	index archive
-v	Verbose
-f	tar archive File to use

Example

In my home directory I have a directory called "`fig`" which contains some layouts I've created with `xfig(1)`. Say I wanted to copy all of the files, with their permissions, owner/group, and directory hierarchy intact. This is where I would use tar:

```
coral:[/home/gunther] $ tar -cvf fig.tar fig
a fig
a fig/machineroom.fig 11 blocks
a fig/machineroom.fig.bak 11 blocks
a fig/workstations.fig 13 blocks
a fig/workstations.fig.bak 13 blocks
a fig/xterms.fig 6 blocks
a fig/xterms.fig.bak 6 blocks
a fig/network.gif 2 blocks
a fig/networkold.fig 14 blocks
a fig/networkold.fig.bak 14 blocks
```

Here I have used `tar(1)` with its verbose flag set, denoted by the '`-v`', so I can see what happens. In this case we're adding (prefix 'a' at each line) to the archive. The archive file is '`fig.tar`', denoted by the '`-f`' flag.

If I wanted to list the contents, I could use the '`-t`' flag:

```
coral:[/home/gunther] $ tar -tvf fig.tar
drwx------ 20250 20250              0 Sep 10 16:32:34 1996 fig/
-rw------- 20250 20250  5197 Sep 04 13:17:06 1996 fig/machineroom.fig
-rw------- 20250 20250  5197 Sep 04 13:17:02 1996 fig/machineroom.fig.bak
-rw------- 20250 20250  6543 Sep 04 15:15:56 1996 fig/workstations.fig
-rw------- 20250 20250  6543 Sep 04 15:10:17 1996 fig/workstations.fig.bak
-rw------- 20250 20250  2931 Sep 04 15:41:17 1996 fig/xterms.fig
-rw------- 20250 20250  2931 Sep 04 15:39:50 1996 fig/xterms.fig.bak
-rw------- 20250 20250        989 Sep 10 12:49:55 1996 fig/network.gif
-rw------- 20250 20250  6915 Sep 10 16:32:34 1996 fig/networkold.fig
-rw------- 20250 20250  6915 Sep 10 16:32:18 1996 fig/networkold.fig.bak
```

This gives you a nice contents listing of all the files in the archive, along with permissions, owner/group ids, file size, modification date, and pathname [2].

To extract the archive, switch the '-c' flag for the '-x' flag:

```
coral:[/home/gunther] $ tar -xvf fig.tar
x fig
x fig/machineroom.fig, 5197 bytes, 11 tape blocks
x fig/machineroom.fig.bak, 5197 bytes, 11 tape blocks
x fig/workstations.fig, 6543 bytes, 13 tape blocks
x fig/workstations.fig.bak, 6543 bytes, 13 tape blocks
x fig/xterms.fig, 2931 bytes, 6 tape blocks
x fig/xterms.fig.bak, 2931 bytes, 6 tape blocks
x fig/network.gif, 989 bytes, 2 tape blocks
x fig/networkold.fig, 6915 bytes, 14 tape blocks
x fig/networkold.fig.bak, 6915 bytes, 14 tape blocks
```

Much like the create archive option, under the verbose flag you get an 'x' at the start of each line telling you the command is eXtracting data.

That's about as difficult as the tar command gets. If you wish to archive a group of files, then the command would be:

```
tar -cvf archive.tar file1 file2 *.c
```

Where you can use explicit filenames (`file1 file2`) or wildcards (`*.c`) or a combination of the both. Of course, the files can themselves be directories.

The archive file doesn't have to be a regular file, it can be (amongst others) a:

- floppy drive, such as `tar -cvf /dev/fd0 *.c`

```
tar -xvf /dev/fd0
```

- tape drive, such as `tar -cvf /dev/rmt0 *.c`

```
tar -xvf /dev/rmt0
```

and so on.

You now have a very easy way to move files around filesystems and around workstations. Just tar the files to file or device, and then extract them where you want them.

Now, as with all things, we want to make our life as easy as possible. If you want to try moving files across filesystems, and you have permissions to write at the destination, try something like the following:

Example:

Moving my `fig` directory from `/home/gunther` to `/tmp`:

```
coral:[/home/gunther] $ cd /tmp
coral:[/tmp] $ ( cd /home/gunther; tar -cvf - fig ) | tar -xvf -
a fig
a fig/machineroom.fig 11 blocks
a fig/machineroom.fig.bak 11 blocks
a fig/workstations.fig 13 blocks
a fig/workstations.fig.bak 13 blocks
x fig
a fig/xterms.fig 6 blocks
a fig/xterms.fig.bak 6 blocks
a fig/network.gif 2 blocks
a fig/networkold.fig 14 blocks
x fig/machineroom.fig, 5197 bytes, 11 tape blocks
a fig/networkold.fig.bak 14 blocks
x fig/machineroom.fig.bak, 5197 bytes, 11 tape blocks
x fig/workstations.fig, 6543 bytes, 13 tape blocks
x fig/workstations.fig.bak, 6543 bytes, 13 tape blocks
x fig/xterms.fig, 2931 bytes, 6 tape blocks
x fig/xterms.fig.bak, 2931 bytes, 6 tape blocks
x fig/network.gif, 989 bytes, 2 tape blocks
x fig/networkold.fig, 6915 bytes, 14 tape blocks
x fig/networkold.fig.bak, 6915 bytes, 14 tape blocks
```

So what have I done? Firstly, I changed directory to /tmp. Now, I didn't want to use absolute pathnames, so I made a complex command (within the parentheses) which first changed directory back to my home, and then issued the tar command. Now, stdout is a file itself, so I have used as my archive destination stdout, represented by '-' in the tar command:

```
(cd /home/gunther; tar -cvf - fig)
```

Now the output of this command will be the tar file. However, all I want to do is reconstruct it in my current working directory (/tmp), so I just need to extract it. I could redirect the above to a file, and then extract the file - why not just redirect it back to stdin and extract that:

```
(cd /home/gunther; tar -cvf - fig) | tar -xvf -
```

In the tar command, using the -x flag, '-' represents stdin.

The following lines of output are just tar(1) showing that it creates (prefix 'a') and extracts (prefix 'x') files. Of course, you can take this one step further. Say you want to transfer files across two machines you have access to, why not use the facilities of the rsh(1) command?

Say, I want to copy files from a machine called emerald, to my machine. I can rsh into emerald, and then tar the files with the output going to stdout and then pick up and extract that archive on my machine.

i.e. `rsh emerald "cd /directory; tar -cvf - files" | tar -xvf -`

Where '/directory' is the location and 'files' the directory or list of files I want to copy.

As you can see, tar(1) makes life a lot easier when it comes to moving large numbers and/or directories. For more information check out the tar(1) man page.

Footnotes:

[1] Under System V you need to use the '-o' flag to ensure the owner of the archive is you, and not the creator. This is important when you un-tar a file you've grabbed off the net.

[2] When creating tar archives, use relative pathnames always. Try to avoid the use of absolute pathnames. The reason, is that when you come to extract an archive which was created using and relative pathname, you can drop it anywhere in a directory tree. With absolute pathnames, it has to go where it was created.

Compare my tar command from above with the following:

```
    coral:[/home/gunther] $ tar -cvf fig-new.tar /home/gunther/fig
    a /home/gunther/fig
    a /home/gunther/fig/machineroom.fig 11 blocks
    a /home/gunther/fig/machineroom.fig.bak 11 blocks
    a /home/gunther/fig/workstations.fig 13 blocks
    a /home/gunther/fig/workstations.fig.bak 13 blocks
    a /home/gunther/fig/xterms.fig 6 blocks
    a /home/gunther/fig/xterms.fig.bak 6 blocks
    a /home/gunther/fig/network.gif 2 blocks
    a /home/gunther/fig/networkold.fig 14 blocks
    a /home/gunther/fig/networkold.fig.bak 14 blocks
```

In this case, you have to restore the files to the exact pathname. Relative pathnames always give you the ability to move things around.

Correction from Last Issue

In the last edition of "Traps & Tricks" we looked at daemons and ports and in doing so mentioned various port numbers. I also managed to pop in a great typo:

> *NNTP - port 113*
>
> *If you get really adventurous, write your own newsreader. Otherwise read articles straight off your newsserver. A silly way to do it - but it does amaze onlookers ;-)*

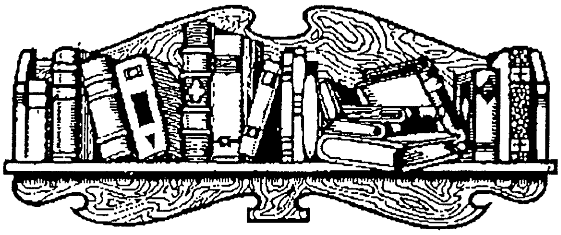
It is of course, port **119**. By the time I realised, the issue was already at the printers. Sorry about that one folks. ♦

Book Reviews

Sub-editor: Frank Crawford

Another AUUGN comes around, and we have a stack of reviews of exciting, and not so exciting books for you. It is easy to tell that the latest and greatest in computing is Java, with three books being reviewed here and more to following in future AUUGNs. However, AUUG is not just Java, but Open Systems and networking of all types, and that is seen in the other reviews, including reviews of books on UNIX internals, network management protocols and other low level details, Perl 5 and even some thoughts on how to protect yourself on the Internet.

You may have noticed, we currently have lots of books coming for review. The current practice is to post a note to the mailing list <auug-books@ansto.gov.au> and the newsgroup aus.org.auug when we have new books available. Unfortunately, this disadvantages members without network connections, or on the end of a low speed link. For people in such a position, either mail, via the AUUG PO Box, or fax me on (02) 9717 9273, with your contact details and preferences.♦



UNIX INTERNALS: THE NEW FRONTIERS

by Uresh Vahalia

Prentice-Hall
1996, 601 pages
ISBN 0-13-101908-2

Reviewed by
Michael Usher
Information Technology Services
University of Sydney
<M.Usher@isu.usyd.edu.au>

Warren Toomey has provided an excellent review of this book in AUUGN Vol. 17, Nr 3, so I won't repeat his praise directly, but rather make a few additional comments. My first impression of the book when I opened the package was *daunting*. The six hundred pages cover a surprising range of topics, often in quite technical detail. I was pleasantly surprised when I discovered the book was actually readable! Vahalia has surmounted what is often presented as a very dry topic and produced an exciting survey of recent operating system design.

There is one minor correction I would make to Toomey's review. He mentions that the book covers a variety of UNIX variants and describes each of these systems. This is not quite true. Vahalia takes a range of topics within operating systems design, and then uses a selection of current UNIX variants to illustrate the various approaches to the problems. In general, he uses SVR4 and Mach as the major exemplars, but other systems get a mention when they have an unusual approach to a problem, notably Digital UNIX, Solaris 2.x and 4BSD. It is true that Linux and the BSD derivatives are missing. Given the 1996 publication date, and that there are only a few references dated 1995, the book must have been substantially complete by early 1995. I don't think the omission devalues the book and I am sure we will see books on the Linux kernel in the near future.

The different approach to the topic means the book has a different audience to the traditional UNIX kernel books. This book is also directly useful as a continuation of Richard Stevens books on UNIX programming. The topics he covers are all areas that programmers and designers need to understand to make best use of the new systems on which they base their products. The sections on multithreading and new filesystems are of special interest here. Description of the different resource allocation strategies is useful to the systems administrator who must manage a variety of UNIX systems or who must choose which system is most suited to a particular application.

As Toomey comments, BSD sockets get only one page at the end of the chapter on STREAMS and even then only as a comparison of the sockets API against TLI under SVR4. This is definitely a major omission, and since the chapter is the last of the book, I can only assume it was not covered due to time constraints.

Overall this is a fine book. There are many detailed diagrams illustrating the various discussions, no noticeable typographic errors, clear and readable descriptions, and the book is simply a joy to read.

I unreservedly recommend this book to those with a technical involvement with modern UNIX systems.♦

TEACH YOURSELF JAVA IN 21 DAYS

by Laura Lemay & Charles L. Perkins
Sams.net Publishing
1996, 527 pages
ISBN 1-57521-030-4

Reviewed by
Peter Gray
Information Technology Services
University of Wollongong
<pdg@uow.edu.au>

If you have not heard of Java by now, you must be living on Mars. You have probably been told that Java is either God's gift to the software industry and the bullet with Microsoft's name on it or an interesting diversion of no real commercial interest. You may have decided to learn Java to see what the fuss is about. If you have, then "teach yourself Java in 21 days" is a good place to start.

The book comes in a rather trendy and attractive purple cover and includes the Java Development Kit from SUN Microsystems for Windows 95/NT and Sparc/Solaris on a CDROM. The book is quite large, 527 pages.

The book claims to be aimed at people who have some programming experience and goes on to say that some BASIC or Pascal at school would be enough. I doubt this very much. I think a few years of C or C++ would be needed to get full benefit from the book. People with little programming experience would find it quite heavy going and may become lost in the object orientated terminology used throughout. Experience with object orientated programming (OOP) would be a distinct advantage.

However, if you do have some programming experience, and have at least a working knowledge of OOP principles, then this book does give an excellent introduction to Java. The material is presented in a well thought out order and the typographic conventions are excellent in helping the reader categorise the information being presented. Each chapter ends with a Q&A section which does address some of the obvious questions a reader may have when reading the text.

The book is divided into three sections, each designed to cover 7 days. The first section (week) introduces the reader to writing Java applications, as opposed to applets. A Java application is much like a program written in any other language. A Java applet is a Java program designed to be loaded into and run by some form of WWW browser. The reader learns about OOP terminology and principles and then how to create Java classes and write small Java applications.

The pace increases in the second section, which is devoted to writing applets. The reader learns about the graphics class, and how to draw onto the screen. Color and fonts are then covered, followed by animation, simple threads, images and sound, then events and user interaction. The section ends by introducing the basic widgets defined in the Java toolkit and briefly mentions networking from within the Java applet. Quite a busy week!

The first two sections were written by Laura Lemay. The last section is written by the other author Charles Perkins and is much more advanced. I doubt any beginning programmer could get through it. Charles Perkins has a quite different style to Laura Lemay and the transition does jar a little.

The section starts by describing modifiers used for controlling class and variable protection, then introduces packages and interfaces. Then three days are spent on exceptions, multithreading and streams respectively. The next day covers native methods and libraries and the last day looks under the hood at the Java virtual machine and bytecodes.

All in all I found the book quite good and would recommend it to experienced programmers wanting to learn Java. I doubt it would be as useful to the inexperienced programmer. I cannot resist making a few comments on Java itself. I like the language a lot. It seems to be a real step forward from C, without introducing the hideous complexity of C++. It can be used just as easily for general purpose applications as for applets. It deserves to succeed and from the amount of interest it is generating, it will.♦

PROGRAMMING WITH JAVA!

by Tim Ritchey
New Riders
1995, 389 pages + CD-ROM
ISBN 1-56205-533-X

Reviewed by
Lawrie Brown
<Lawrie.Brown@adfa.oz.au>

Java is one of the super-hyped areas vying for attention at present, so not surprisingly we see a raft of material appearing which attempt to fulfill the perceived demand. This book, is one such. In brief, I have to see that my reaction to it is one of disappointment and frustration. This book seemed to promise a lot, but frankly, it doesn't deliver. At first glance it suggested that it would teach you how to program in Java, however on closer inspection the title "Programming with Java" is actually apt. It spends a lot of time telling you what Java is, how to

Notification of Change

AUUG Inc

UNIX® AND OPEN SYSTEMS USERS

You can help us! If you have changed your mailing address, phone, title, or any other contact information, please keep us updated. Complete the following information and either fax it to the AUUG Membership Secretary on (02) 9332-4066 or post it to:

AUUG Membership Secretary
P.O. Box 366
Kensington, NSW 2033
Australia

(Please allow at least 4 weeks for the change of address to take effect..)

□ The following changes are for my personal details, member #:________________

□ The following changes are for our Institutional Member, primary contact.

□ The following changes are for our Institutional Member, representative 1.

□ The following changes are for our Institutional Member, representative 2.

PLEASE PRINT YOUR OLD CONTACT INFORMATION (OR ATTACH A MAILING LABEL):

Name/Contact: ________________________________
Position/Title: ________________________________
Company: ________________________________
Address: ________________________________
________________________ Postcode ________
Tel: BH ________________ AH ________________
Fax: BH ________________ AH ________________
email address: ________________________________

PLEASE PRINT YOUR NEW CONTACT INFORMATION:

Name/Contact: ________________________________
Position/Title: ________________________________
Company: ________________________________
Address: ________________________________
________________________ Postcode ________
Tel: BH ________________ AH ________________
Fax: BH ________________ AH ________________
email address: ________________________________

AUUG Secretariat Use

Date: ________________
Initial: ________________
Date processed: ________________
Membership # ________________

MEMBERSHIP APPLICATION

INSTITUTIONAL

UNIX® AND OPEN SYSTEMS USERS

To apply for AUUG membership, complete this form and return it with payment in Australian dollars to:

REPLY PAID 66, AUUG MEMBERSHIP SECRETARY,
P.O. BOX 366, KENSINGTON, NSW 2033, AUSTRALIA
Tel: +61 2 9361-5994 or 1 800 625 655 • Fax: +61 2 9332-4066

Tick this box if you wish your name withheld from mailing lists made available to vendors. □

NOTE: Please do not send purchase orders - perhaps your purchasing department will consider this form to be an invoice. Foreign applicants please send a bank draft drawn on an Australian bank.

We, ______________________________
(Company Name)

do hereby apply for:

Renewal/New* Inst. membership of AUUG □ $350.00
International air mail □ $120.00

TOTAL REMITTED AUD$________
(Cheque, money order, or credit card)

I/We agree that this membership will be subject to the rules and by-laws of AUUG as in force from time to time, and that this membership will run from time of joining/renewal until the end of the calendar or financial year.

I/We understand that I/we will receive two copies of the AUUG newsletter, and may send two representatives to AUUG sponsored events at member rates, though I/we will have only one vote in AUUG elections, and other ballots as required.

Signed______________________Date ________
Title ______________________________

INSTITUTIONAL MEMBER DETAILS:
To be completed by institutional members only.

Following are our specified contacts. The primary contact holds the full member voting rights. The two designated reps will also be given membership rates to AUUG activities including chapter activities. By default a regional chapter will be selected for you. If you would rather nominate a chapter, please specify in space provided (indicate NONE for no chapter). (Please print clearly or type)

Primary Contact______________________________
Position/Title ______________________________
Address ______________________________
______________________________Postcode__________
Bus. Tel: ________________ Bus. Fax:______________
e-mail address ______________________________
Local Chapter Pref. ______________________________

1st Rep.______________________________
Position/Title ______________________________
Address ______________________________
____
Bus. Tel: ________________ Bus. Fax:__________
e-mail Address ______________________________
Local Chapter Pref. ______________________________

2nd Rep.______________________________
Position/Title ______________________________
Address ______________________________
____
Bus. Tel: ________________ Bus. Fax:__________
e-mail address ______________________________
Local Chapter Pref. ______________________________

Please charge $____________ to my
□ Bankcard, □ Visa, □ Mastercard
Account number:______________________________
Expiry date:______________________________
Name on card:______________________________
Signature:______________________________

AUUG Secretariat Use

*Chq: bank*______________ *bsb* ______________
a/c ______________ *#* ______________
*Date:*______________ *$* ______________
*Initial:*______________
*Date processed:*______________
*Membership #*______________

AUUG Inc. as a user group, exists to provide UNIX® and open systems users with relevant and practical information, services, and education through cooperation among users.

MEMBERSHIP APPLICATION
INDIVIDUAL

AUUG Inc
UNIX® AND OPEN SYSTEMS USERS

To apply for AUUG membership, complete this form and return it with payment in Australian dollars to:

REPLY PAID 66, AUUG MEMBERSHIP SECRETARY,
P.O. BOX 366, KENSINGTON, NSW 2033, AUSTRALIA
Tel: +61 2 9361-5994 or 1 800 625 655 • Fax: +61 2 9332-4066

Tick this box if you wish your name withheld from mailing lists made available to vendors. □

NOTE: Please do not send purchase orders - perhaps your purchasing department will consider this form to be an invoice. Foreign applicants please send a bank draft drawn on an Australian bank.

INDIVIDUAL OR STUDENT MEMBERSHIP:

I, ______________________________ do hereby apply for:

Renewal/New membership of AUUG □ $ 90.00
Renewal/New Student membership □ $ 25.00 (please complete certification portion)
International air mail □ $ 60.00

TOTAL REMITTED: AUD$________ (Cheque, or money order, or credit card)

I agree that this membership will be subject to the rules and by-laws of AUUG as in force from time to time, and that this membership will run from time of joining/renewal until the end of the calendar or financial year.

____
Signature

____
Date

LOCAL CHAPTER DESIGNATE:

You can participate in the activities of a local AUUG Chapter. Part of your fee will be given to the chapter to support those activities. By default a regional chapter will be selected for you. If you would rather nominate a chapter, please specify here ____________________ (indicate NONE for no chapter).

TO BETTER SERVE YOU, PLEASE PRINT YOUR CONTACT INFORMATION:

Name/Contact: ______________________________
Position/Title: ______________________________
Company: ______________________________
Address: ______________________________
______________________________Postcode__________
Tel: BH____________________ AH ____________________
Fax: BH____________________ AH ____________________
email address:______________________________

Over for Institutional Membership

STUDENT MEMBER CERTIFICATION: *(to be completed by a member of the academic staff)*

I, ______________________________ certify that
(administrator)
______________________________is a full time student at
(name)
______________________________ and is expected to
(institution)
graduate approximately ____________________.
(date)

____
Signature

____________________ ____________________
Title *Date*

Please charge $______________ to my
□ Bankcard, □ Visa, □ Mastercard
Account number:______________________________
Expiry date:______________________________
Name on card:______________________________
Signature:______________________________

AUUG Secretariat Use

*Chq: bank*______________ *bsb* ______________
a/c ______________ *#* ______________
Date: ______________ *$* ______________
Initial: ______________
Date processed: ______________
Membership # ______________

AUUG Inc. as a user group, exists to provide UNIX® and open systems users with relevant and practical information, services, and education through cooperation among users.

TAME THE WILD WEB

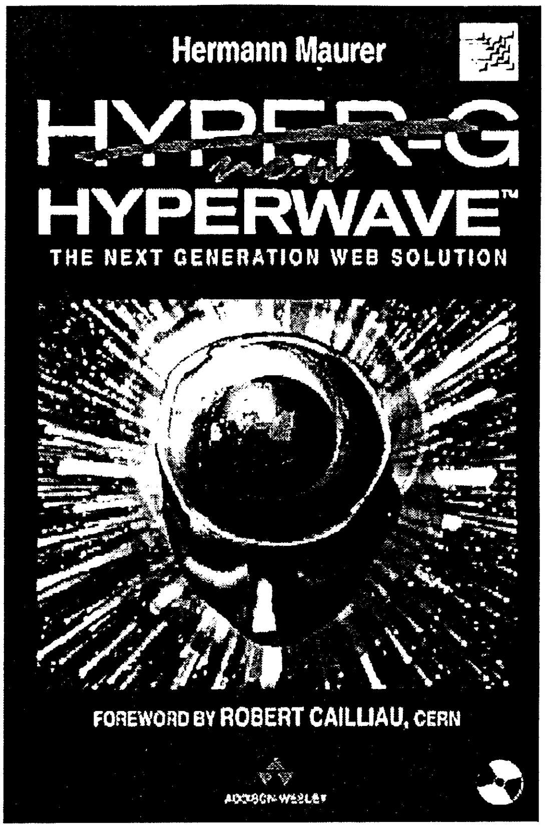
HYPERWAVE

THE NEXT GENERATION WEB SOLUTION

HERMANN MAURER   0 201 40346 3

SPECIAL AUUG PRICE: $59.45

(RRP $69.95)


Hermann Maurer
HYPER-G
now
HYPERWAVE™
THE NEXT GENERATION WEB SOLUTION
FOREWORD BY ROBERT CAILLIAU, CERN
ADDISON-WESLEY

Written by the team that created Hyper-G and its commercial version HyperWave, this book is the definitive reference for the next generation hypermedia system for the Internet. If you have a large, complex and highly dynamic information space, and/or if you have, or are seeking to create, a high quality multi-author environment - read this book!

The CD-Rom includes an evaluation copy of the HyperWave server, the native UNIX and MS-Windows clients as well as the HyperWave administration tools.

Hyper-G was conceived and developed at the IICM of Graz University of Technology in Australia by a team headed by Hermann Maurer and Frank Kappe, both prolific and widely published authors.

"FINALLY, SOME SORELY NEEDED STRUCTURE FOR THE WILD WEB."

- JAKOB NIELSEN, SUN MICROSYSTEMS DISTINGUISHED ENGINEER, USA.

15% DISCOUNT FOR ALL AUUG MEMBERS

ORDERING

I would like to order ___ copy/copies of HYPERWAVE at the special AUUG price of $59,45 ($69.95 less 15%) = $_______

AUUG Membership no: ________

___ I enclose a cheque for $_________ OR

___ Please charge my credit card   __ Bankcard   __ Visa   __ Mastercard

Card No ____ ____ ____ ____   Exp date ________

Name ____

Title & Organisation ______________________________

Address ____

State ______________ Postcode __________ email ______________

Phone ____________________ Fax ______________________

RETURN TO:
Trade Marketing Coordinator
Addison Wesley Longman
95 Coventry Street
South Melbourne 3205
Ph: 03 9697 0664
Fax: 03 9699 2041

Addison
Wesley
Longman

If you have any questions, please email the AUUG Chapter committee (`cauug.committee@auug.org.au`), or the Internet Project Management Committee (`ipmc@pcug.org.au`), or contact myself or another committee member.

Jeremy Bishop

News from the NSW Chapter

NSW Chapter has monthly meetings on the third Thursday of each month upstairs at the Occidental Hotel, York St Wynyard.

August meeting

The August meeting was held on the fourth Thursday and had three features. The first feature was Frank Crawford who spoke about SAGE-AU Conference which was held in Brisbane in July. Not many NSW members had made the trek to Queensland for the Conference and were keen to find out what had taken place.

Frank spoke in some detail about the papers presented by Dinah McNutt from the US. Dinah's methods of managing large numbers of systems contained something useful for everyone. Frank then covered the other talks at the conference, including such topics as the migration from windows 3.11 to NT, Certification and Authentication methods, UNIX Security, on-line regulation, Auscert and event management, amongst other topics. Frank's talk was well received, talking about a conference sounds dry but he brought it to life, and there was discussion on a range of issues raised during the conference.

The second feature was a stump the experts/question and answer session. Members asked questions of the gathering, and I think we did quite well, we could answer questions on the intricacies of Windows 95, ufsdump on Solaris 2.5 failing in certain hardware, the merits of different revision control software, and cheap effective help desk systems. The stumper was "I boot my DEC alpha from the install cdrom to restore root but the boot cd kernel does not include tape devices, what can I do?" "Install linux", someone suggested.

The third feature was the after meeting drinks and dinner, catching up with old friends and meeting new members.

Farewell Bernie!

Bernie Goodheart has recently resigned as president of the NSW Chapter of AUUG. Bernie is returning to the UK after 8 years in Australia. Thank you Bernie for your efforts over the last 18 months as President; it's been fun. We look forward to seeing you out here for future Summer Conferences.

Upcoming meetings

September

NSW will not hold a meeting in September as this would be the same week as the AUUG winter conference.

October

Our October meeting will have John Terpstra talking about the latest developments in SAMBA.

November

Our November meeting will be by Bruce Howarth from the University of Technology in Sydney who will talk about advances in performance management and capacity planning.

We look forward to seeing you come along and support the local chapter meetings.

Contacts

Wayne Bell
President
`wbell@sydney.dialix.oz.au`

David Purdue
Secretary
`david.purdue@aus.sun.com`

auug-nsw-committee
`auug-nsw-exec@auug.org.au`

Canberra Chapter: September 96 newsletter

DATES OF FUTURE MEETINGS:

8 Oct and 12 Nov at Open Solutions Centre

AUUG Inc. - Canberra Chapter - September 1996 newsletter

SEPTEMBER MONTHLY MEETING

When:
7:30 pm for 8:00 pm, Tuesday, 10 September 1996

Where:
Open Solutions Centre, 15 Barry Drive

Speaker:
Tony Shepherd

Topic:
Identifying and Responding to Information Security Risks

ABSTRACT

Computer intruders are individuals that range from the curious to the those who maliciously damage your system. These people are not innocents that are "just having fun". They are criminals whose actions are costly in terms of computer down time, extra work for system administrators, and the stress it often causes those involved. Computer intruders can be members of staff that attack the system from within, or external perpetrators that try and gain unauthorized access from the external network. Anyone who has a computer network connected to the world, whether via the Internet, modem banks or X.25, is at risk.

This presentation will discuss how to assess your risk and to determine where the real risk lies. The talk will discuss a number of different kinds of attacks that are commonly reported to AUSCERT, ranging from the curious probe to a fully blown compromised network involving sites world wide. It will also cover procedures that should be followed in the event of an unauthorized intrusion, and how sites can recover and use the experience to decrease the chance of further attacks. Techniques and procedures that are useful in detecting an intrusion will be highlighted, as will some tools that can be used to lessen the chance of a successful attack.

BIO

Tony Shepherd is a member of the Australian Computer Emergency Response Team (AUSCERT) based in the capital of Queensland, Brisbane. He gained First Class Honours at the University of Queensland with a double major in Computer Science and Mathematics. Tony has four years experience in the information technology industry, specialising in computer systems security and system administration. His experiences range from assisting sites with computer security issues and analysis of computer security vulnerabilities, to the implementation of computer security systems designed to prevent unauthorised intrusions. Tony's experience within AUSCERT places him in a unique position to analyse and present trends in computer security incidents on the Internet.

REMINDER!

A reminder to all AUUG members who are also members of the PCUG/AUUG Internet Project (commonly known as TIP) that the revised Acceptable Use Policy (AUP) comes into effect from 1st October 1996, and that you MUST sign a declaration that you accept the Internet Project's Acceptable Use Policy by that date, or your access to TIP will be suspended.

The AUP details what is and isn't allowed with regard to use of TIP, and sets out the rights of users and the actions that can be taken upon breach of these terms and conditions. The AUP has been revised on the basis of legal advice, and is designed to protect your rights as a user of the system, and of other users.

Declarations need to be signed by a person over 18 years of age, and in the presence of an AUUG committee member, a PCUG committee member, or a PCUG Staffer (at the PCUG Centre - see below).

Note that if you do not sign the Declaration by 30th Sept 1996, your TIP account will be suspended until we hear from you. If you do not wish to accept the revised AUP, please let us know. TIP will refund the purchase price of any unused Advanced Access time allocation you may have (less the minimum usage for the period since 1st Feb 1996), and will close your TIP account.

Copies of the Acceptable Use Policy and the Declaration form will be available at the Chapter Monthly Meetings (second Tuesday of each month), and are also available at the PCUG Centre, 3/17 Dalby Street, Fyshwick, open 9am-5pm each Saturday and Sunday.

From the Western Front

Edited by Tom Hallam <thallam@geol.uwa.edu.au>

Meeting Information

WAUG meets at the Freeway Hotel, 55 Mill Point Road, South Perth. We meet at 6:15pm on the **third Wednesday of each month.**

Our meetings are advertised in the Diary column of the Computers section of Tuesday's *West Australian.*

If you need further information about the next meeting, please contact Mark or one of the committee.

SPEAKERS ARE NEEDED, especially ones who can actually commit to giving a talk on a certain date! So if you can give a talk, or know someone who can, please let us know. Mark (our meeting organiser) cannot produce them out of thin air. SAGE also needs speakers (See Local SAGE-AU below)

WAUG Email Aliases, Newsgroups and Web Page

WAUG has the following mail aliases on uniwa.uwa.edu.au:

`waug-membership`
for membership enquires

`waug-chair`
our Chairperson

`waug-meetings`
our meeting organiser

`waug-secretary`
our Secretary

`waug-newsletter`
for newsletter contributions or enquires

`waug`
for general correspondence (will be read by the Secretary, as a paper letter would be).

So, for example, you may send general correspondence to `waug@uniwa.uwa.edu.au.`

Check us out on the World Wide Web at:
`http://www.auug.org.au/auug/waug/waug.html`
(thanks Canberra AUUG).

Also see the newsgroups `wa.waug` and `aus.org.auug` for announcements and discussion.

Committee Contact Details

Chair:	Don Griffith 351 7691 `griffith@cs.curtin.edu.au`
Vice-chair:	Luigi Cantoni 474 3700 `lui@DIALix.oz.au`
Secretary:	Tom Hallam 380 2665 `thallam@geol.uwa.edu.au` (AUUGN Sub-editor)
Treasurer:	Patrick Ko 483 8111 `pko@DIALix.oz.au`
Meeting Organiser:	Mark Baker 491 6081 `baker@telecomwa.oz.au`
Ordinary Committee:	Daniel Baldoni `flint@cs.curtin.edu.au` (Meeting Reporter)
	David Buck `dbuck@ncc.telecomwa.oz.au`
	Glenn Huxtable 328 8288 `glenn@fs.com.au`
	Peter Wemm

For Systems Administrators: Local SAGE-AU Meetings

The WA Regional Group of the Systems Administrators Guild of Australia (SAGE-AU)

meets on the **First Tuesday of each month at 6pm, in room G3 at the Alexander Library** If you manage computer systems for a living, we'd like to have you along.

SAGE-AU is **NOT** another UNIX group. All systems and network administrators are welcome. We would particularly like to see more PC network administrators attending, so if you know any, send them along. I'd like to see lots of Novell, NT, OS2 and MAC people attending.

For more information, please contact Don Griffiths `<griffith@cs.curtin.edu.au>`, (09) 351 7691 or myself, Tom Hallam `<thallam@geol.uwa.edu.au>`, (09) 380 2665. For information about SAGE-AU in general, you may also look at `ftp://ftp.sage-au.org.au/pub/SAGE-AU` and `http://www.sage-au.org.au:8080/`.

AUUG BOOK CLUB
&
PRENTICE HALL AUSTRALIA

20% DISCOUNT TO AUUG MEMBERS

Please send me a copy/copies of the following book —

□	Red Hat Linux Unleashed ISBN 0672309629	Binder	$89.95	$71.95
□	Apache Server Unleashed ISBN 1575211475	Paper	$89.95	$71.95
□	Running a Perfect Web Site with Apache ISBN 0789707454	Binder	$84.95	$67.95
□	Web Server Handbook ISBN 013239930X	Paper	$79.95	$63.95
□	Netware to Internet Gateways ISBN 0135217741	Binder	$89.95	$71.95
□	Building Linux Internet Server ISBN 1562055259	Paper	$64.95	$51.95
□	Building a Unix Internet Server ISBN 1562054945	Binder	$54.95	$43.95

AUUG members receive 20% discount below recommended retail price
Please send me the selected book/s on 30-Day approval

Name: ____________________ Position: ____________

Position: ____________________

Address: ____________________

____________________ Telephone: ____________

PLEASE SEND MY BOOK/S ON 30-DAY APPROVAL

□ Enclosed cheque for $ _______ (Payable to'Prentice Hall Australia')

□ Charge to me **OR** □ Company purchase Order No. ________

□ Please charge my: □ Bankcard □ Visa □ MasterCard

Expiry Date: ____________ Credit Card No: □□□□ □□□□ □□□□ □□□□

Signature: ____________________

□ Please add my name to your mailing list in the following areas: *(please list subject areas)*

____

____

____

____

____

Mail or fax completed order form to Prentice Hall Australia,
PO Box 151, Brookvale NSW 2100

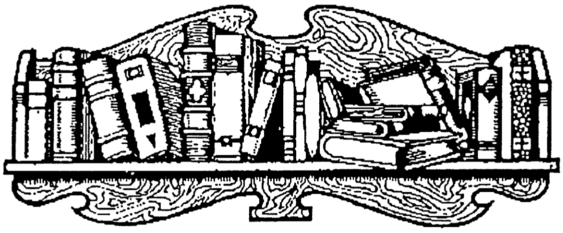
OR ✆ Use our PHONE SERVICE by calling
Jan Blenkinsop SYDNEY (02) 9907 5693

PRENTICE HALL AUSTRALIA
7 Grosvenor Place, Brookvale NSW 2100
Tel: (02) 9939 1333 • Fax: (02) 9905 7934 A.C.N. 000 383 406

have heard of the Internet, but I would think that their experience with Internet would be very new.

Australian Internet consumers need to take care of buying internationally. Australian laws would apply to Australian sellers, but not necessarily to international sellers. Unless you have lost a fair fraction of a million dollars, I would not even begin to consider litigating a company located in another country. But, you can cancel credit card transactions for goods not received or wrongly charged through the local credit card organisation.

The book is readable and has useful information for people who participate in the Internet, but its utility depends on the reader and how Internet "street smart" they are.♦



PERL 5 DESKTOP REFERENCE

by Johan Vormans
O'Reilly and Associates
1996, 39 pp., $9.95
ISBN 1-56592-187-9

Reviewed by
Frank Crawford
ANSTO
<frank@ansto.gov.au>

This is a mini-review for a mini-book. The *Perl 5 Desktop Reference* is basically an update of the reference card that is distributed with the Perl Reference Manual. Unfortunately, the changes that occurred from Perl 4 to Perl 5 meant that the previous reference card was missing many important features.

This has now been corrected, and the new card lists the syntax and all the functions now found in Perl 5, including both arguments and return values. While it is no longer possible to include all the details on a "card", the booklet itself, is not very large, being slightly smaller in dimensions than the card.

The value of such a booklet cannot be over estimated, as, for simple queries, it is much more convenient than either a manual or online help. From discussions with WoodsLane, apparently many others agree, as this is one of their biggest selling items at the moment.

If you are using Perl 5 for anything (or even still using Perl 4), this booklet will be of invaluable assistance, and you should race out and get a copy.♦

BANDITS ON THE INFORMATION SUPERHIGHWAY

by Daniel J. Barrett
O'Reilly & Associates Inc.
1996, 246 pp
ISBN 1-56592-156-9

Reviewed by
Alex Kowalenko
<alex@mpx.com.au>

Crime and computers have always gone together. Since the earliest James Bond book to the Internet, the perception is there, that computers can aid the rouge gangster to achieve his goals. I might be generalising here, but the target has been always been other banks, other corporations, other countries. In a word, someone else.

Many books are available about protecting our computers and their precious contents from rouge hackers. Stories about famous hackers circulate in the popular press. Methods of constructing firewalls, corporate security standards, and encryption are available, but what about protecting ourselves?

In our modern cities people have codes of conduct that protect us from people with undesirable motives. We lock our cars and homes, we are careful where we walk alone, and we are careful what we say and display to other people about ourselves. We have a sense of "street smarts" that guide our behaviour.

"But the Internet is different!" cries the newbie surfer. "It's one big happy global village!" Is it? Who are the members of the of this new village? Aren't they the same members of this old unfashionable planet. Then why do we think that everyone would have adopted the ethics of this new global village?

Daniel Barrett has written a book on this subject. He goes through various situations people have found themselves in. These vary from meeting people on the Internet, buying and selling on the Internet, business scams and issues involving privacy.

For instance, many of us would not think of placing a list of our home's contents outside our home for everyone to read. We don't even leave our curtains wide open so that everyone can see what we have. It's an advertisement to burglars. But people are willing to list a catalogue their collection of records in their home page, boast about their new computer in newsgroups, list their expensive hobbies next to their home address on their online resume.

For people who have been using the Internet for some time, they would have heard about some the stories that Daniel lists in his book, or heard, or even experienced similar stories. But for new starters, it's just one big new cyber world, without any of problems in the current world we live in.

I would recommend the book to new starters on the Internet, it lists some of the problems that people can get themselves into if they are not careful. But for people who have been using the Internet for a while, they would have developed their own Internet "street smarts". Reading the book would confirm some of their ideas, but also there are things that you can learn. But, I don't know whether I would pull out the pull-out card "15 ways to spot a Internet Bandit" and stick it to my monitor.

One feature that devalues the book in Australian eyes is that it discusses American law, and list American consumer protection agencies. No doubt Australian consumer protection agencies would the network management in general. But as a first reading on either SNMP or network management, the book is hardly suitable. The novice reader would better start from something like Chapter 25 of TCP/IP Illustrated, Vol. 1 by W R Stevens or a similar material and use the Simple Book as a second reading.

At the end, another word of caution. Throughout the book, the author ridicules the OSI approach to network management. Though I largely agree with his arguments, the way of total rejection is a bit 'religious' and should be treated with caution.

Overall, the Simple Book is a good quality reading. For those who are well prepared to tackle it, the book provides a wealth of details and the information that is very hard to find anywhere else.♦

Enterprise Networking: Data Link Subnetworks

by James Martin, Kathleen Kavanagh Chapman
& Joe Leben
Prentice Hall New Jersey
1996, 431 pp
ISBN 0-13-507575-0

Reviewed by
Peter Lane
<peterl@acslink.net.au>

My first mistake was to take this book to work, to "read during lunch". It was immediately seconded by a colleague who is studying a TAFE level IT/IM certificate, and just happened to be doing Data Communications this term. The book promptly disappeared for two weeks, and was returned to my desk with a small note. Apparently, my colleague would like to have kept the book for longer and they had found it reasonably useful, especially the sections describing different network topology's. Finally, they recommended the book for "new students".

So I started reading. Frankly, I found it pretty tedious. This surprised me, as my first impressions on first pulling it out of the package and having a quick scan through were that it promised to be a first-class presentation. In fairness, however, the book appears to be designed as a reference volume rather than a cover-to-cover read, and serves that purpose quite well. By this stage, I'd managed to work out why the sub-title; Data Link Networks. This is a companion volume to another volume entitled *Enterprise Networking: Strategies and Transport Protocols.* When I first received Data Link Subnetworks, I was very keen on obtaining the Transport volume as well; by the time I'd finished it I wasn't so keen any more.

The organisation of the volume is quite logical. There are 25 chapters organised as four parts, and four Appendices.

Part I is titled *Fundamentals,* and covers the expected material - definitions, the OSI Reference model (surprise! surprise!), interactions between the Transport and Data Link layers, architectures, applications and the Transport Layer in brief.

Part II deals with the underlying technologies that form the foundation of the Data Link layer, covering Network Driver software, Physical transmission mechanisms, repeaters and hubs, and basic protocols and protocol linkages.

Part III covers WAN technologies, specifically Leased Line type circuits, X.25, ISDN, Frame Relay, ATM and Wireless WAN links. Each technology is described precisely and concisely, showing how each differs. The analysis is particularly focused on the packet structure and the Data Link protocols.

Part IV does the same thing for Local Area technologies - Ethernet, Token Ring, Token Bus and ARCnet, FDDI, Apple LocalTalk and Wireless LAN (wireless uplinks - not pervasive networks). An additional chapter covers the details of the Metropolitan Area network standards.

The Appendices comprise an annotated listing of Standards Organisations (which I probably would have found useful as a student), a description of the OSI Reference Model, a description of the IEEE/ISO/ANSI LAN Architecture, and a glossary.

Which brings us back to the recommendation of my colleague, in my first paragraph. Despite the back cover marketing blurb, this book is primarily suited for student reference use. Secondarily, it could well serve as a reference for a communications professional who specialised in a particular area of the field, but occasionally needed a reasonably detailed reference for other sub-specialities. Cynically, I believe that Prentice Hall may well be relying on the previous reputation of one author to attract buyers for this volume. Certainly, in my opinion the volume could have been easily compiled by any reasonable competent researcher. However, the finished result certainly merits a place on your bookshelves if either of the above conditions applies; and if you can cost-justify it!♦

as a reference book for a computer professional, it is less than ideal. With the plethora of books on Java now available in the bookstores, there must be other texts which are superior reference works for Java.♦

THE SIMPLE BOOK: AN INTRODUCTION TO NETWORK MANAGEMENT

by Marshall T. Rose
Prentice Hall PTR
1996, 294 pp, CD-ROM included, hardcover
ISBN 0-13-451659-1

Reviewed by
Michael Haldey
<mhaldey@ozemail.com.au>

The Simple Book is a book about the Internet Network Management Framework (written by one of its designers), specified by the Simple Network Management Protocol (SNMP). The second revised edition of the Simple Book reflects the current condition of the framework specified in the SNMP version 2 documentation.

The author concentrates on the conceptual and descriptive sides of the specification. The implementation issues which were present in the first edition were removed from this edition. A reference·to another book of the same author is given for those who are interested.

The Simple Book consists of six chapters and three appendices.

The first Chapter is a brief introduction into the book.

The second Chapter is an introduction into the framework outlining the main concepts and the model behind SNMP. The sections present the overviews of the network management protocol (traversal, traps, transport mappings), the management information (managed objects, proxy relationships), and the data representation (this section is basically an introduction to ANS.1).

Chapter 3 deals with the management information. It is divided into four sections, describing the structure of the management information (formal descriptions using ASN.1), MIB modules, conformance statements, and coexistence.

Chapter 4 describes the Administration model which is supposed to define authentication, privacy and authorisation policies. As the current SNMP framework still doesn't do it properly, the author just concentrates on the terms, tasks and definitions. The Chapter is fairly short.

Chapter 5 examines the protocol operations in the management framework. Overall, it is the most detailed and useful chapter, as it explains how all the pieces are working together. The author starts from discussing the styles of the interaction between SNMP entities, then move to the descriptions of the various requests and operators. Special attention is paid to the conceptional rows ('tables'). The chapter also presents the Transport Mapping with various protocols, and obviously, BER and serialisation (transforming ASN.1 defined information into the octet strings).

The last Chapter, which is more rather conceptional than technical discusses some existing problems as well as possible future extensions of the framework.

The Appendices provide a lot of additional information on the Internet standards and documents, related RFCs and the other resources for further studies of the theory and implementations of the framework. A few implementations are provided on the attached CD-ROM.

Overall, the book is the most complete description of the Internet Network Management Framework available today. It is very detailed with many formal descriptions. The author's writing style is quite good; he is even trying to make some fun out of the often boring material. Numerous 'soapboxes' informally present some 'behind the scenes' information, background material, historic issues, etc. Throughout the book, a lot of attention is paid to further development trends and SNMPv1 compatibility issues.

This quality presentation still doesn't mean that the book is 'simple' or even easy to read. It looks like the author tries to achieve two not very compatible goals - make the book 'simple' and provide a lot of details and formal descriptions. In the Simple Book, the second goal was achieved, but the formality and an amount of formal ASN.1 descriptions has overloaded the text with details and made the overall picture very hard to see. Unfortunately, the introductory Chapter 2 is not well-structured, with few examples. You wouldn't find a lot of schemas either. (Another 'wish-list' item - why not to put related RFCs to the CD-ROM?)

So, despite the title, the Simple Book is not really 'simple'. It is the best for advanced readers who have some understanding of the subject and need more details about the framework. These readers can consider the book an introduction to the network management as well. A lot of background material would allow a reader to get some understanding of create an environment to create and use it, even providing quite a comprehensive overview of the language features. But when it comes to the crunch, when you actually want to write something and think, well how do I do this, the answer's just not there! Very annoying!

In a little more detail, the book contains 14 chapters and 2 small appendices. The first 4 chapters introduce the concepts of interactive content and Java, using Java applets and applications, and a brief overview on installing the Java Developers Kit (JDC). It is not until chapter 5 that any Java code is introduced, and then it is the usual "Hello World!" application and applet.

Chapters 6 through 13 then provide an overview of the various language features. These cover the basic language constructs (data types, expressions, and control structures), the concept of classes (and interfaces and packages), threads and exceptions, the various standard Java and HotJava class libraries, and finally interfacing with native code. Each of these chapters include a number of code fragments and example programs illustrating the concepts being discussed. These are good as far as they go. The author also tries to sketch the range of entities available and how to use them, but with insufficient detail for this to be a suitable reference. I felt it gave me some idea of the language and library capabilities, but not sufficient knowledge to branch out and write code of any depth.

The final chapter 14 introduces the Java Virtual Machine, the virtual architecture that Java programs are compiled down to and run on. Again, whilst introducing the various ideas, there is a frustrating lack of detail, just when things look like getting interesting.

The appendices briefly provide a guide to other sources of information, and briefly sketches the components in the AWT window library used by Java applets.

The book also includes a CD-ROM. This contains an (inevitably superseded) copy of the Java Developers Kit for Win 3.1, 95, NT, and for Solaris 2.x. It also contains a number of sample applets, and copies of the code examples from the book. Virtually all of the contents can be trolled relatively easily off the net, so there's nothing critical here, useful though it may be to some to have it collected together.

In summary, this book disappointed me by not living up to its promise. For someone with a background as a competent C/UNIX programmer like mine, this is not the book you want. Perhaps for someone interested more in the management perspective, wanting to know what the Java phenomena is about, and to get some idea of its capabilities without getting lost in the details, this may well be appropriate - it does have some good conceptual introductions. To be fair to the author, the book deals with a very large field, and clearly seems focused as an introductory book. Unfortunately that means the depth of detail was sacrificed. So consequently, in my opinion, its coverage is just too superficial to make this a suitable a book for technical experts.♦

PRESENTING JAVA: AN INTRODUCTION TO JAVA AND HOTJAVA

by John December
Sams.net Publishing
1995, 207 pp
ISBN 1-57521-039-8

Reviewed by
Peter Williams
<Peter.Williams@dsto.defence.gov.au>

This book is aimed at the casual reader rather than the accomplished computer user. Consequently, approximately half of the book consists of hype about the potential of Java with respect to the World Wide Web which would have been interesting had it not been so repetitive. The difference in content between Section One (Understanding the Potential of Java and the Web) and Section Two (Exploring Java's Potential) was insufficient to hold this reviewers interest. In addition, the repetition of information from chapter to chapter within each of these sections was high enough make them a very boring read.

Section Three of this book gives a brief but informative account of how to obtain and use the HotJava browser. As the author states, the information in this section will quickly become dated due to the rapid development of HotJava. Nevertheless, this section would prove useful in getting HotJava up and running.

Section Four of this book gives a fairly shallow but thorough description of the Java language. As for the previous section, the information in this section will become dated as Java evolves but still provides a good starting point. The book has three appendices (A: Further Information Sources, B: Java Language Reference and C: HTML Tag References) which nicely encapsulate most of the useful information contained in the book.

In summary, this book may appeal to the casual reader who wishes to gain a better understanding of Java's potential contribution to the Web, However,